

Jaakko Pöllänen

VANERITEHTAAN LIIMAKEITTIÖN TIEDONKERUU

Opinnäytetyö
Sähkötekniikan koulutusohjelma


Toukokuu 2015




MAMK

University of Applied Sciences

KUVAILEHTI

 MAMK University of Applied Sciences		Opinnäytetyön päivämäärä 26.5.2015	
Tekijä(t) Jaakko Pöllänen		Koulutusohjelma ja suuntautuminen Sähkötekniikan koulutusohjelma	
Nimeke Vaneritehtaan liimakeittiön tiedonkeruu			
Tiivistelmä <p>Tämän opinnäytetyön tarkoituksena oli toteuttaa tiedonkeruu UPM Savonlinnan vaneritehtaan liimakeittiön sekoittajalle yksi. Samalla myös liimakeittiön valvomo uusittiin. Liimakeittiön tehtävä on tuottaa vanerin ladonnan tarvitsema liima.</p> <p>Tiedonkeruu toteutettiin InTouch-ohjelmalla. Ennen tiedonkeruuta suunnittelin, toteutin ja käyttöönotin uuden valvomon. Valvomo oli aiemmin toteutettu Visual Basicilla. Tiedonkeruu tehtiin InTouchilla, joten samalla myös valvomo päätettiin uusida InTouch-valvomoksi. Liimakeittiön yhteisiä toimintoja ja sekoittajaa yksi ohjataan Siemensin S7 315-2 DP -logiikalla. Tiedonkeruun ja valvomon uusimisen myötä myös logiikkaohjelmaa täytyi muokata.</p> <p>Tiedonkeruun ohjelmointi ja käyttöönotto onnistui hyvin. Valvomon suunnittelu vei paljon aikaa, mutta käyttöönotto onnistui suunnitellusti. Valvomon käyttö helpottui uusimisen myötä, sillä vanhan valvomon layout ei ollut kaikilta osin paikkaansa pitävä. Myös liimakomponenttien kulutuksen seuranta on tarkentunut tiedonkeruun myötä.</p>			
Asiasanat (avainsanat) Tiedonkeruu, liimakeittiö, valvomo			
Sivumäärä 35+1	Kieli Suomi	URN	
Huomautus (huomautukset liitteistä)			
Ohjaavan opettajan nimi Hannu Honkanen		Opinnäytetyön toimeksiantaja UPM-Kymmene Oyj, Savonlinnan vaneritehdas	

DESCRIPTION

 <div style="display: inline-block; vertical-align: middle;"> <div style="font-size: 2em; font-weight: bold; margin: 0;">MAMK</div> <div style="font-size: 0.8em; margin: 0;">University of Applied Sciences</div> </div>		Date of the bachelor's thesis 26.5.2015	
Author(s) Jaakko Pöllänen		Degree programme and option Electrical engineering	
Name of the bachelor's thesis A plywood mill glue kitchen production data collection			
Abstract <p>The purpose of this Bachelor's thesis was to implement production data collection for mixer one in the glue kitchen of Savonlinna plywood mill. In the same project human- machine interface of glue kitchen was modernized. The glue needed for plywood bonding is prepared in the glue kitchen.</p> <p>First a new human machine interface was made with InTouch software. Before project it was made with Visual Basic. The original job was to make production data collection of glue kitchen with InTouch software. Production data collection demanded changes to InTouch program and to logic program. Logic sends production data to InTouch. For that I had to make code to logic. InTouch makes csv file to which production data is written. Csv file is written with InTouch scripts. These scripts exist but I needed to make changes to them. Server retrieves csv file from local computer and makes production data collection report.</p> <p>As a result of the work glue kitchen have new human machine interface and mixer 1 have production data collection.</p>			
Subject headings, (keywords) Production data collection, glue kitchen, HMI			
Pages 35+1	Language Finnish	URN	
Remarks, notes on appendices			
Tutor Hannu Honkanen		Bachelor's thesis assigned by UPM-Kymmene Oyj, Savonlinna plywood mill	

SISÄLTÖ

1	JOHDANTO	1
2	UPM	1
2.1	UPM Plywood.....	2
2.2	Savonlinnan vaneritehdas	2
3	LIIMAKEITTIÖ	3
4	VALMISTELEVAT TYÖT	4
4.1	Logiikan liittäminen verkkoon.....	5
4.2	Todellisen layoutin selvittäminen	7
5	WONDERWARE INTOUCH -VALVOMO	8
5.1	Valvomon layoutin suunnittelu.....	8
5.1.1	Päänäyttö	10
5.1.2	Harts- ja pesuvesisäiliöt ja kovetesiilo.....	11
5.1.3	Asetukset	11
5.1.4	Muut sivut	12
5.2	Logiikan ja valvomon kommunikointi.....	12
5.2.1	Yhteyden valvonta	12
5.2.2	Käytetyt datatyypit.....	14
5.3	Tilatiedot logiikalta valvomolle.....	14
5.3.1	Hälytykset.....	15
5.3.2	Sekoittimen tilatieto	17
5.4	Ohjaukset ja asetukset logiikalle.....	18
5.5	Reseptit	19
5.5.1	Reseptien hallinta InTouchissa	20
5.5.2	Reseptien käyttö logiikassa	22
6	TIEDONKERUU	23
6.1	Tiedonkeruumäärittelyt	23
6.2	Tiedonkeruun ohjelmointi	23
6.2.1	Ohjelmointi logiikassa	24
6.2.2	Ohjelmointi InTouchissa	28
6.3	Tiedonkeruuraaportti.....	31
7	YHTEENVETO	34

LIITE

1 Tiedonkeruumäärittelyt

1 JOHDANTO

Valitsin opinnäytetyön aiheeksi Savonlinnan vaneritehtaan liimakeittiön tiedonkeruun toteuttamisen. Tehtaamme liimakeittiössä on kaksi erillistä liimansekoittajaa sekä niille yhteiset varastosäiliöt. Opinnäytetyön tarkoituksena oli saada sekoittaja yksi tiedonkeruuseen. Liimakeittiössä tehdään vaneritehtaan ladonnan tarvitsema liima. Liima on merkittävä kuluerä vanerin valmistuksessa, joten liimakomponenttien kulutusta haluttiin seurata tarkemmin. Tiedonkeruusta saadaan tarkat tiedot liimakomponenttien todellisesta kulutuksesta, kun aiemmin kulutusta voitiin seurata ainoastaan liimakomponenttien toimitustietojen mukaan.

Tiedonkeruut ovat tehty tehtaallamme InTouch-ohjelmalla, joten tiedonkeruun teon yhteydessä myös liimakeittiön valvomo muutettiin InTouch-valvomoksi. Aiemmin valvomo oli tehty Microsoftin Visual Basicilla. Kyseessä oli tehtaan ainut Visual Basic valvomo, joten se päätettiin korvata tehtaalla yleisesti käytetyllä InTouchilla.

Liimakeittiö on siirretty Savonlinnan vaneritehtaalte toiselta tehtaalta, ja siirron yhteydessä valvomoa ei muokattu. Tämän seurauksena valvomon layout oli vain osittain todellisuutta vastaava, joka on vaikeuttanut liimakeittiön käyttöä. Myös valvomon puutteet puolsivat valvomon uudistamista.

2 UPM

UPM yhdistää bioteollisuutta ja perinteistä metsäteollisuutta. UPM:llä on tuotantolaitoksia 14:sta maassa. Lisäksi myyntiverkosto on maailmanlaajuinen. UPM:n palveluksessa on noin 21 000 työntekijää ja vuoden 2013 liikevaihto oli noin 10,1 miljardia euroa. /1, s.1./

Suomessa UPM:llä on 27 tuotantolaitosta. Henkilöstöä on noin 8110. Vuonna 2013 liikevaihto Suomessa oli noin 4,3 miljardia euroa. /1, s.30./

UPM jakautuu kuuteen liiketoiminta-alueeseen: UPM Biorefining, UPM Paper Asia, UPM Paper ENA (Eurooppa ja Pohjois-Amerikka), UPM Energy, UPM Raflatac ja UPM Plywood /1, s.1-2/.

UPM syntyi syksyllä 1995, kun Kymmene Oy ja Repola Oy, sekä sen tytäryhtiö Yhtyneet Paperitehtaat ilmoittivat yhdistymisestään. Uusi yhtiö, UPM-Kymmene, aloitti toimintansa 1.5.1996. UPM:llä on Suomessa pitkät perinteet metsäteollisuudessa. Konsernin ensimmäiset puuhiomot ja paperitehtaat sekä sahalaitekset käynnistyivät 1870-luvun alkupuolella. Sellunvalmistus aloitettiin 1880-luvulla ja paperin jalostus 1920-luvulla. Vanerin valmistukseen konsernissa ryhdyttiin 1930-luvulla. /2./

2.1 UPM Plywood

UPM Plywood tuottaa vaneria ja viilua pääasiassa rakentamiseen, kuljetusvälineisiin, sekä muuhun teollisuustuotantoon. UPM on johtava vanerintoimittaja Euroopassa. Tuotantolaitoksia on Suomessa, Virossa ja Venäjällä. /1, s.2./

Vanerin ja viilun tuotantokapasiteetti on yhteensä noin miljoona kuutiometriä vuodessa. UPM:n vaneri ja viilutuotteita myydään pääasiassa WISA-tavaramerkillä. Uutta lämpömuotoiltavaa vaneria myydään UPM Grada -tavaramerkillä. /1, s.23./

2.2 Savonlinnan vaneritehdas

Savonlinnan vaneritehdas tuottaa koivuvaneria kuljetusväliteollisuuteen, betonointiin ja muihin teollisiin loppukäyttökohteisiin. WISA-vanereita tehdään myös maxikokoisina levyinä. /3./

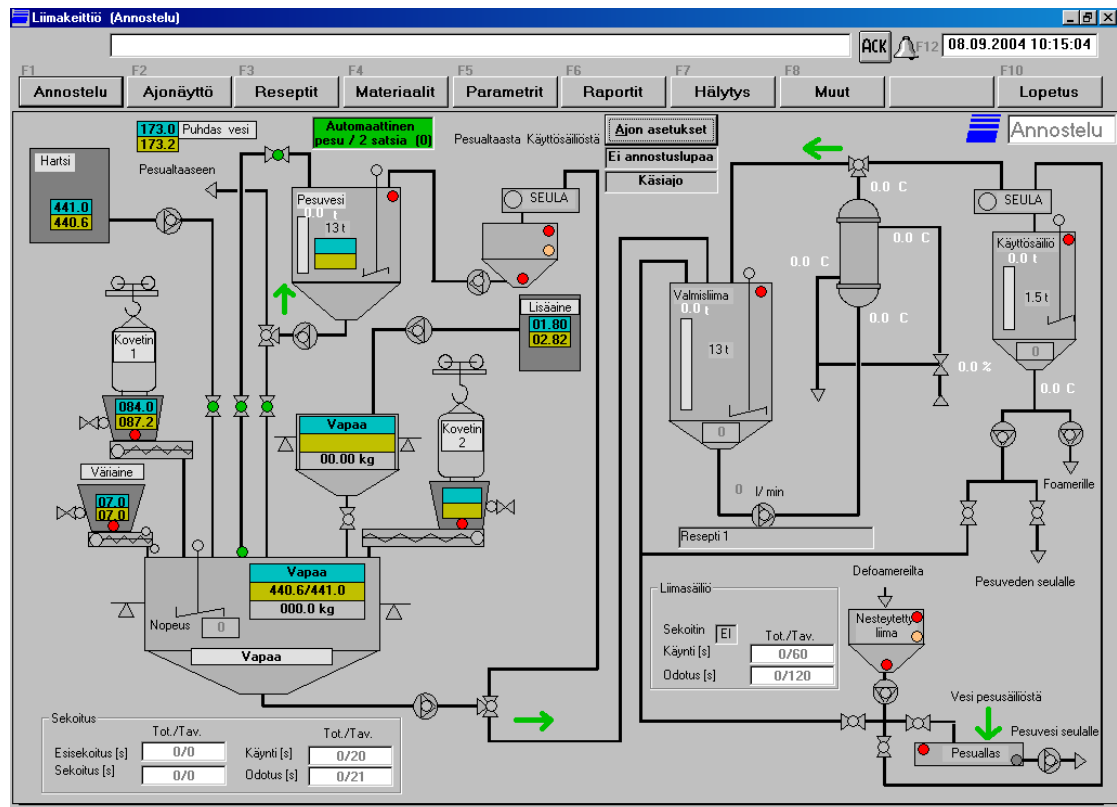
Tehtaan kapasiteetti on 100 000m³ koivuvaneria vuodessa. Tehtaalla on henkilöstöä noin 270. /3./

3 LIIMAKEITTIÖ

Liimakeittiön tehtävä on tuottaa vanerin ladonnan tarvitsema liima. Savonlinnan vaneritehtaalla on neljä ladonta-asemaa. Liimakeittiössä on kaksi liiman sekoitinta. Liiman sekoittimessa sekoitetaan liiman komponentit keskenään. Tehtaallamme käytetty liima koostuu fenolihartsista, kovetteesta ja vedestä. Liiman valmistukseen voidaan käyttää myös ladonnasta tulevaa pesuvettä. Lisäksi liimaan lisätään tehdaskohtaista tunnistusväriainetta. Sekoittimissa on vaakavat, joiden avulla liiman komponentit saadaan yksittelen annosteltua. Fenolihartsille ja pesuvedelle on omat varastosäiliöt. Myös väriaineelle on pienet sekoitin kohtaiset annostelusäiliöt, vesi tulee vesijohtoverkosta ja kovete kovetesäiliöstä välisäiliön kautta. 1. sekoittajaan voidaan annostella kovetetta myös säikeistä.

Valmis liima siirretään sekoitin kohtaisilla siirtopumpuilla päivä säiliöön. Päiväsäiliöitä tehtaallamme on kaksi, mutta yleensä vain toinen on kerrallaan käytössä. Päiväsäiliöistä liima siirretään kiertopumpulla viulun ladontaan, ja kierrosta ylijäävä liima palautuu takaisin päivä säiliöön.

Tällä hetkellä liimakeittiön sekoittajaa yksi, sekä liimakeittiön yhteisiä toimintoja ohjataan Siemensin S7-300 -sarjan logiikalla. Käyttöliittymänä toimii Visual Basicilla toteutettu valvomo, joka on liitetty profibus protokollaa käyttäen logiikkaan. Sekoittajan yksi vaakapäätteenä on Raute Precisionin WB-951. Vaakapääte kommunikoi logiikan kanssa käyttäen profibusprotokollaa. Toinen vastaava vaakapääte on lisäainevaakapääte, joka ei ole käytössä. Laitteisto on tullut tehtaallemme käytettynä, ja aiempi kokoonpano ei täysin vastaa nykytilannetta. Tämän vuoksi valvomossa on paljon ylimääräisiä, vääränlaisia tai väärässä paikassa olevia komponentteja. Kuvassa 1 näkymä vanhasta valvomosta.



KUVA 1. Liimakeittiön vanhan valvomon annostelu -sivu

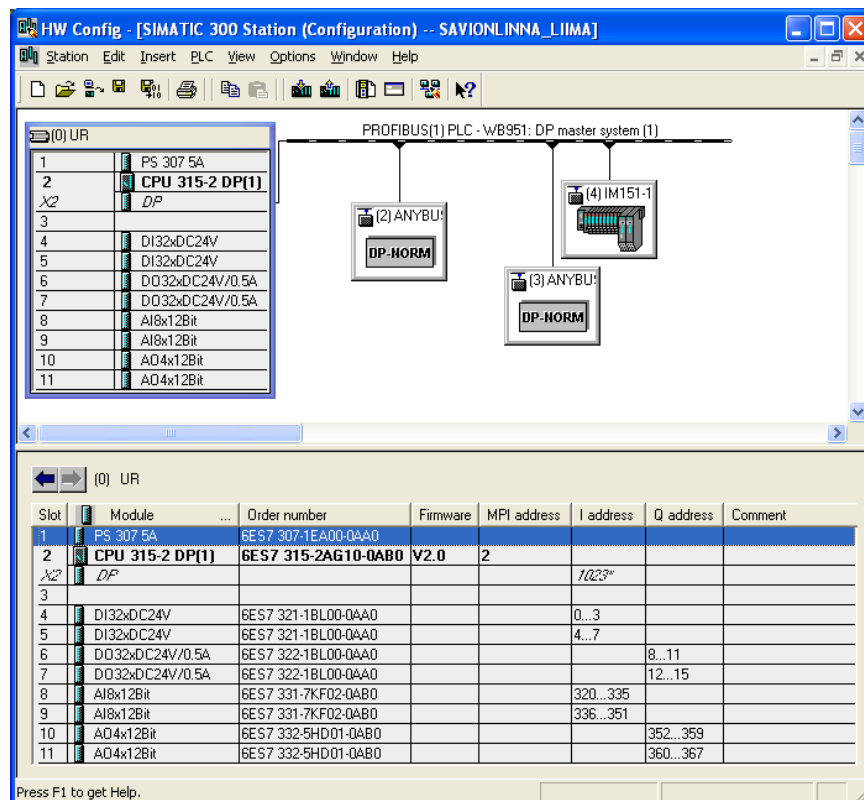
Sekoitinta kaksi ohjataan vanhemmalla Raute Precision WB-950 -vaakapäätteellä. Tällä päätteellä ei pelkästään punnita, vaan myös ohjataan sekoittimeen liittyviä toimilaitteita. Sekoittimella ei ole PC-pohjaista valvomoa, vaan ohjaukset ja asetukset suoritetaan kytkimillä ja vaakapääteen näyttöpaneelilla.

4 VALMISTELEVAT TYÖT

Ennen varsinaisen työn aloittamista täytyi selvittää ohjausten nykytilanne sekä varmistaa, löytyykö liimakeittiön logiikkaan ohjelmaa. Täsmälleen paikkaansa pitävää ohjelmaa ei löytynyt, mutta kahta eri ohjelmaa yhdistelemällä sai lähes paikkaansa pitävän ohjelman kommentteineen ja hardware-määrittelyineen. Kun logiikkaohjelma oli saatu ajantasalle, lisäsin logiikkaan verkkokortin, sillä uuden valvomon ja tiedonkeruun vuoksi logiikan verkkoliityntä on välttämätön. Lisäksi liimakeittiön todellinen layout tuli selvittää, jotta valvomosta saa todellisuutta vastaavan.

4.1 Logiikan liittäminen verkkoon

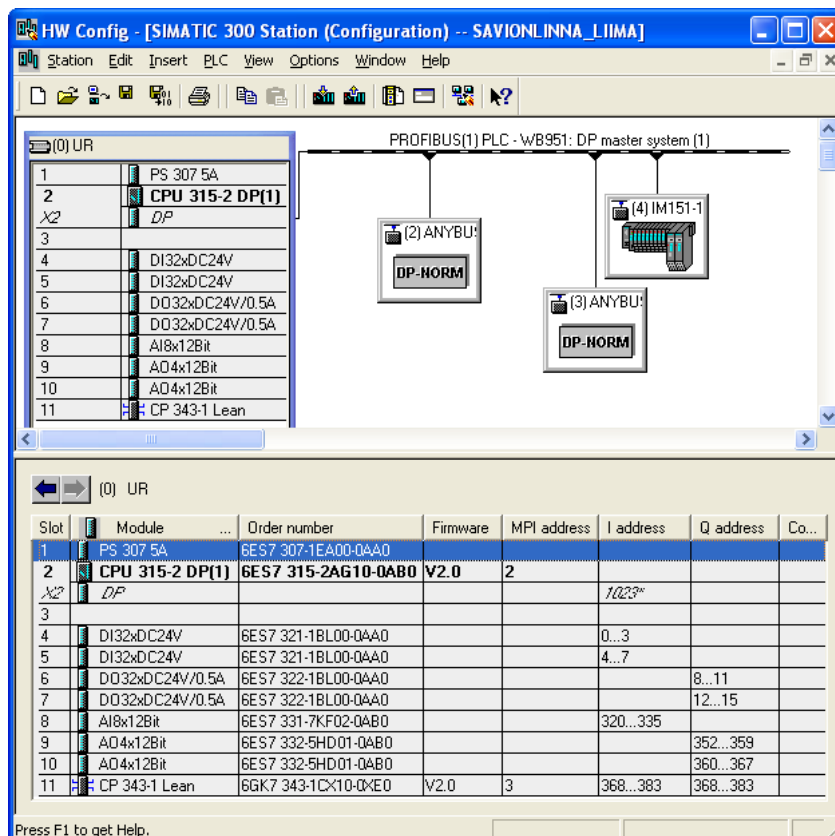
Kun paikkaansa pitävä logiikkaohjelma oli olemassa, päätin seuraavaksi liittää logiikan verkkoon. Logiikka oli välttämätöntä saada verkkoon, sillä tulevan InTouch-valvomon on tarkoitus kommunikoida logiikan kanssa ethernet verkon kautta. Lisäksi verkkokortin asennus mahdollisti logiikan tilojen monitoroinnin ja muutosten teon etänä. Logiikan liittämiseksi verkkoon löytyi varastosta sopiva, käytetty CP-343-1 Lean -kortti. Linjalla olevaan 315-2 DP -logiikkaan ei voinut lisätä nykyistä enempää kortteja, joten yksi kortti täytyi poistaa. Valitsin poistettavaksi analogia tulokortin, sillä analogiatuloja jäi kortin poistamisen jälkeen vielä viisi vapaaksi. Kuvassa 2 alkuperäinen hardware-konfiguraatio.



KUVA 2. Alkuperäinen hardware-konfiguraatio

Hardware-muutos tuli suorittaa mahdollisimman nopeasti ja tuotantoa häiritsemättä. Ongelmana oli, että hardware-muutosten ajaksi logiikka menee stop-tilaan, eikä täten ohjaa mitään. Lisäksi korttien fyysisten asennusten ajaksi oli katkaistava logiikan ja korttien jännitteensyöttö. Tuotannon kannalta tärkeintä oli, että liimankiertopumppu pyörii koko ajan, joten päädyin laittamaan liimankiertopumpun taajuusmuuttajan käsinajolle muutoksen ajaksi.

Logiikan hardware-muutokset tehdään Simatic Managerin HW Configissa. Ensin poistin kortin yhdeksän AI8x12Bit. Poistaminen tapahtui valitsemalla kortti yhdeksän ja painamalla delónäppäintä ja tämän jälkeen hyväksymällä kortin poisto. Kun kortti oli poistettu, täytyi siirtää analogia lähtökortit AO4x12Bit yhtä pykälää ylemmän eli paikkoihin yhdeksän ja kymmenen. Kortit siirrettiin öraahamallaö ne hiirellä yksitellen yhtä korttipaikkaa ylemmäksi. Jotta analogialähtökorttien osoitteet sai pysymään ennallaan, täytyi siirtoa ennen ottaa kortin osoiteasetuksista pois öSystem defaultö ö asetus (Object properties→Addresses). Kun korttipaikka 11 oli saatu vapaaksi, lisättiin siihen CP-343-1 Lean ökortti. Lisääminen tapahtui klikkaamalla hiiren oikeaa nappia korttipaikan 11 kohdalla ja valitsemalla öInsert objectö. Tämän jälkeen kortti löytyi valitsemalla SIMATIC 300 → CP300 → Industrial Ethernet → CP 343-1 Lean → 6GK7 343-1CX10-0XE0 → V2.0. Kun verkkokortti oli lisätty, muutettiin verkkokortin asetukset oikeiksi. Valittiin verkkokortin Object Properties, jonka General ö välilehdeltä valittiin Properties. Properties sivun välilehdeltä Parameters muutettiin IP address ja Subnet mask halutuiksi. Tämän jälkeen samalta välilehdeltä luotiin uusi Subnet New öpainikkeella. Verkon nimi jätettiin oletuksena olevaksi Ethernet(1) ja painettiin OK. Kun asetukset olivat oikein, siirryttiin takaisin HW Configin perusnäkykymään ja valittiin ylävalikosta Station → Save and Compile. Kuvassa 3 muutettu hardware-konfiguraatio.



KUVA 3. Muutettu hardware-konfiguraatio

Hardware oli nyt valmiina logiikkaan latausta varten, joten liimankiertopumpun taa-juusmuuttaja laitettiin käsinajolle ja katkaistiin logiikalta ja korteilta syöttösähköt. Kun sähköt oli katkaistu, poistettiin analogiatulokortti paikasta yhdeksän ja siirrettiin analogialähtökortit paikoista 10 ja 11 paikkoihin yhdeksän ja kymmenen. Tämän jälkeen lisättiin verkkokortti paikkaan 11 ja kytkettiin syöttösähköt takaisin päälle. Kun logiikalla oli sähköt, ladattiin uusi hardware-määrittely logiikkaan valitsemalla ylävalikon PLC:n alta Download... . Tämän jälkeen ohjelma pyytää valitsemaan halutun moduulin. Tässä tapauksessa ei ole kuin yksi valittavissa oleva moduuli, joten painetaan OK. Tämän jälkeen aukeaa ikkuna öSelect Node Addressö, josta voidaan valita käytettävä MPI-address, sekä varmistaa vielä CPU-moduulin tyyppi. MPI-osoitteeksi ehdotetaan osoitetta kaksi, joka käy, joten painetaan OKópainiketta. Hardware-konfiguraation lataaminen logiikkaan alkaa, ja hetken päästä aukeaa Stop Target Modules-ikkuna, jossa ohjelma ehdottaa logiikan asettamista stop-tilaan, jotta hardware-konfiguraatio voidaan ladata logiikkaan kokonaisuudessaan. Painetaan OKópainiketta, jolloin logiikka menee stop-tilaan ja lataaminen suoritetaan loppuun. Kun lataus on valmis, aukeaa Downloadóikkuna, jossa ilmoitetaan, että CPUómoduuli on stop-tilassa ja kysytään, haluatko käynnistää moduulin uudestaan. Painetaan YES-painiketta, jolloin logiikka siirtyy RUN-tilaan.

4.2 Todellisen layoutin selvittäminen

Todellinen layout oli tärkeää selvittää tarkasti ennen valvomon suunnittelun aloittamista. Liimakeittiöstä ei ollut paikkaansa pitävää kaaviota, joten ainoa tapa layoutin selvittämiseen oli seurata putkia ja annosteluruuveja ja tutkia, mistä mihin ne menevät ja mitä toimilaitteita väliltä löytyy. Kaikkien varastosäiliöiden tilavuudet eivät myöskään olleet tiedossa. Kovetesiilosta löytyi rakennekuva, josta selvisi sen tilavuus, mutta päiväsäiliöiden, pesuvesisäiliön ja hartsisäiliöiden tilavuudet piti selvittää mittamalla ja laskemalla.

5 WONDERWARE INTOUCH -VALVOMO

Wonderware-ohjelmistoja on asennettu maailmanlaajuisesti jo yli 750 000 työasemaan 150 000 eri kohteessa kaikilla merkittävillä teollisuuden aloilla. Wonderware-ohjelmistot liittävät eri valmistajien automaatio- ja IT -laitteet yhdeksi helposti hallittavaksi kokonaisuudeksi: Rockwell, Siemens, Schneider, ABB, Honeywell, SAP, Microsoft jne. Wonderware-ohjelmistolla toteutetaan prosessi- ja koneautomaation, raportoinnin, erilaiset ylläpidon ja huollon ratkaisut sekä MES-tuotannonohjauksen.

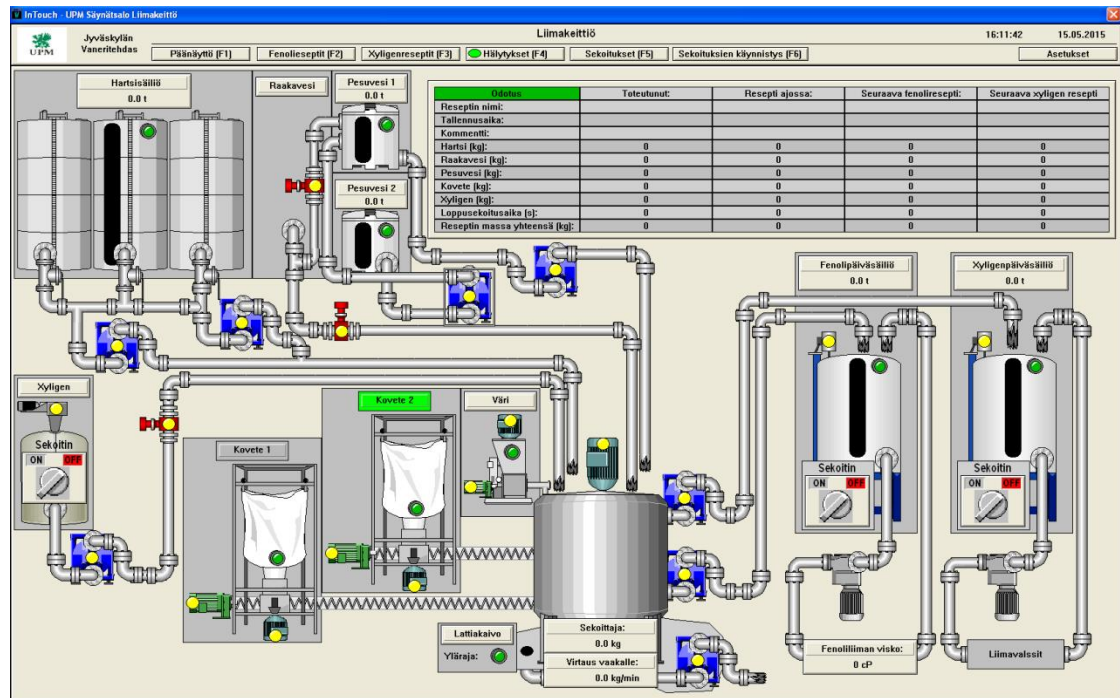
/4./

Wonderware InTouch on UPM Plywoodilla yleisesti käytössä oleva valvomo-ohjelma. Sitä käytetään linjojen ohjauksien lisäksi tiedonkeruuseen. Liimakeittiön valvomo toteutettiin InTouchin versiolla 9.5, joka on julkaistu jo vuonna 2005.

InTouchia muokattaessa ja testattaessa tarvitaan kahta ohjelmaa. WindowViewer on ohjelma, jossa valvomoa käytetään. WindowMaker on puolestaan ohjelma, jolla ohjelmoidaan valvomoa. InTouchin ohjelmoinnissa käytin apuna InTouch Reference Guidea, Recipe Manager User's Guidea ja InTouch HMI Scripting and Logic Guidea.

5.1 Valvomon layoutin suunnittelu

Aloitin valvomon tekemisen layoutin suunnittelulla. Layoutin suunnittelun pohjana oli toiselta tehtaalta saatu liimakeittiön InTouch-valvomo sekä tehtaallamme ollut vanha Visual Basic -valvomo. Kumpikaan ei sellaisenaan sopinut tehtaamme liimakeittiöön. Päädyin muokkaamaan toiselta tehtaalta saatua InTouch-valvomoa, jotta se saataisiin vastaamaan liimakeittiötämme. Kuvassa 4 näkymä alkutilanteesta.



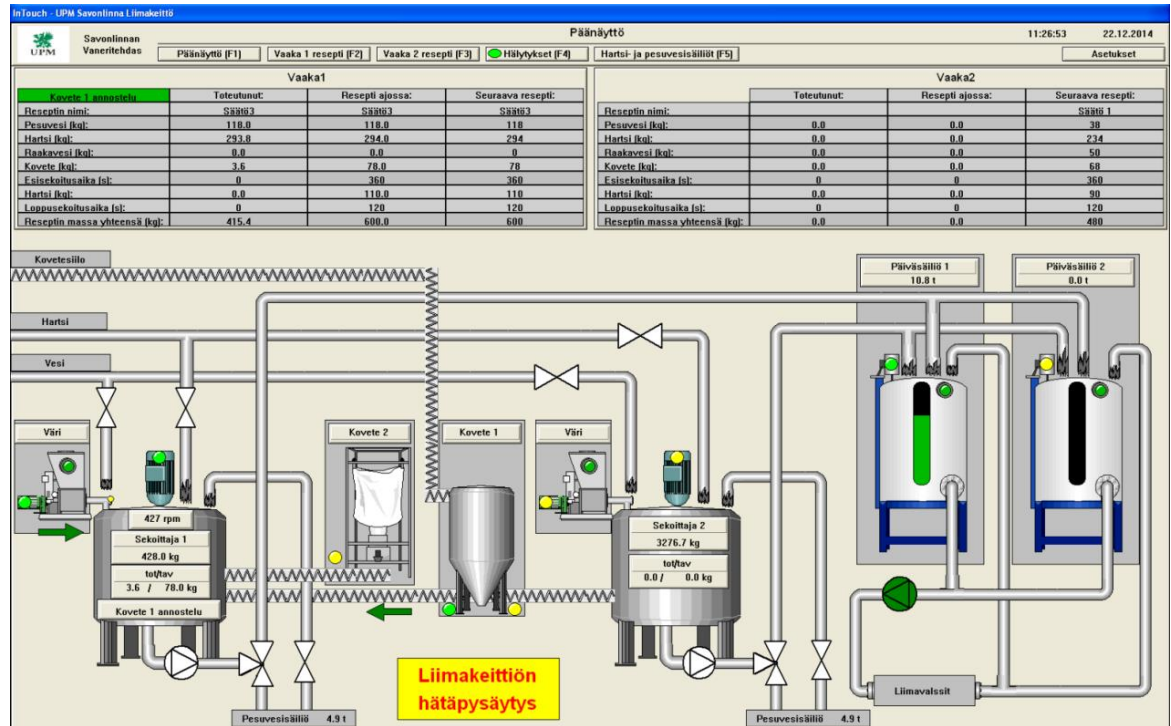
KUVA 4. Valvomon pohja toiselta tehtaalta

Valvomon sivujen suunnittelu oli yllättävän paljon aikaa vievä työvaihe. Lopulta päädyin ratkaisuun, jossa valvomon rakenne muodostuu seuraavista sivuista: päänäyttö, harts- ja pesuvesisäiliöt ja kovetesiilo, hälytykset ja asetukset. Sekoitinta kaksi ei vielä tämän opinnäytetyön yhteydessä lisätä samaan logiikkaan muiden ohjauksien kanssa, mutta valvomon sivut tehdään sekoitin kaksi huomioiden. Myös logiikkaohjelmassa otetaan huomioon toisen sekoittimen lisäys järjestelmään.

Venttiilejä, moottoreita ja pumppuja sisältävillä sivuilla on niiden toiminnalle samantyyppiset indikoinnit. Hahmottamisen helpottamiseksi venttiilit vaihtavat värinsä valkoisesta vihreäksi kun ne ovat auki ohjattuna. Pumput ovat valkoisia ollessaan seis ja vihreitä pyöriessään. Sekoittimien ja syöttöruuvien moottoreiden toimintaa indikoivat merkkivalot, jotka ovat moottoreiden ollessa seis keltaiset, ja moottorien pyöriessä vihreät. Lisäksi aineita siirrettäessä näytetään siirtosuunnan osoittavia vihreitä nuolia.

5.1.1 Päänäyttö

Päänäyttö on käyttäjän kannalta merkittävin näyttö, ja siltä käyttäjän tulee nopeasti nähdä mikä tilanne liimakeittiöllä on. Tämän vuoksi pyrin pitämään näytön mahdollisimman yksinkertaisena, mutta samalla myös tarpeeksi kattavana kokonaisuuden hahmottamiseksi. Kuvassa 5 liimakeittiön päänäyttö.

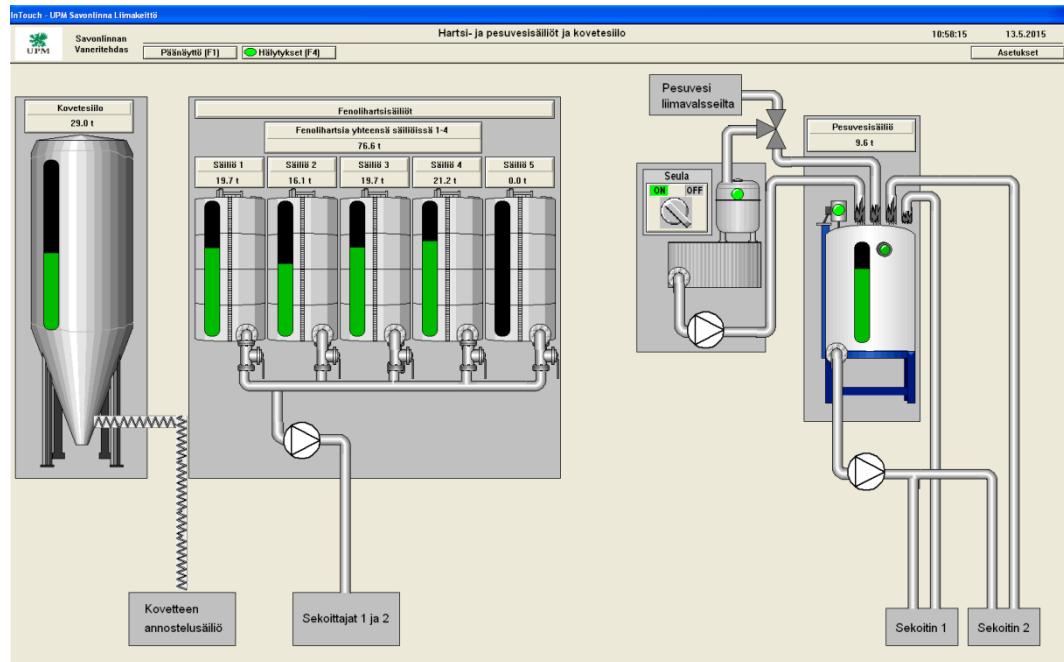


KUVA 5. Liimakeittiön valvomon päänäyttö

Tärkeintä oli saada liiman sekoittajat näytölle. Lisäksi päiväsäiliöissä olevan liiman määrä on erittäin oleellinen tieto käyttäjille. Kummallekin sekoittajalle tuli oma taulukko ruudun yläreunaan, josta näkee, mikä resepti on ajossa, missä vaiheessa panoksen teko on ja ovatko tavoite ja toteutuneet määrät samat. Taulokosta näkee myös seuraavaksi ajettavan reseptin. Sekoitinsäiliöissä on neljä tilinäyttöä: sekoittajan kierrosnopeus, vaakana bruttopaino, annosteltavan aineen toteutunut ja tavoite massa, sekä sekoittajan tilinäyttö. Päiväsäiliöiden pinnankorkeus ilmaistaan graafisesti ja pinnan korkeuden mukaan laskettu massa numeerisesti. Käytettävän reseptin valintaan vaikuttaa merkittävästi pesuvien määrä, joten lisäsin pesuvesisäiliön pinnankorkeuden mukaan lasketun massan päänäytölle.

5.1.2 Harts- ja pesuvesisäiliöt ja kovetesiilo

Sivu on lähinnä varastosäiliöiden tilatietojen seuraamista varten. Sivulla näkyy kovetesiilo, fenolihartsisäiliöt, pesuvesisäiliö, sekä pesuvesiseula. Sivulta on pääsy myös sivuille päänäyttö, hälytykset ja asetukset. Kuvassa 6 kuva sivusta.



KUVA 6. Harts- ja pesuvesisäiliöt ja kovetesiilo-sivu

Sivulla näkyy kovetesiilo, jonka vaa'an lukema ilmoitetaan numeerisena ja massan mukaan laskettu pinnankorkeus graafisena. Hartsisäiliöiden ja pesuvesisäiliön pinnanmittaukset on esitetty graafisesti sekä pinnankorkeuden mukaan laskettu massa numeerisesti. Sivulta löytyy myös pesuvesiseulan ohjaus. Liimavalssseilta tulevan pesuveden venttiili ei ole tällä hetkellä käytössä, mutta venttiili on olemassa, joten se on piirretty sivulle.

5.1.3 Asetukset

Asetukset-sivulta saadaan asetettua liimakeittiön tärkeimmät asetukset. Sivulta on myös pääsy materiaaliparametrit- ja tiedonkeruuasetukset sivuille. Edellä mainittujen sivujen asetuksia ei käyttäjän tarvitse normaalisti muuttaa.

Asetukset-sivulta löytyvät asetukset kummallekin sekoittajalle, sekä liimakeittiön yhteisille laitteille. Asetuksista voi muuttaa mm. sekoittajien ajotapaa, vaaka tyhjä rajaa sekä pesuasetuksia. Päiväsäiliöiden asetuksista voi muuttaa tilausrajoja ja sekoittimien

nopeudet sekä päiväsailiön kaksi osalta jaksottaisen käynnin sykliä. Pesuvesisäiliön sekoittajan käynnistys ja liimankiertopumpun käynnistys ja nopeuden säätö löytyvät myös asetukset-sivulta.

5.1.4 Muut sivut

Edellä mainittujen sivujen lisäksi valvomosta löytyy kummankin vaa'an reseptisivut, hälytykset sivu, säiliöiden trendisivut ja laitteiden käsinajosivut. Reseptisivuilta voidaan vaihtaa, muokata, luoda tai poistaa sekoittimien reseptejä sekä muokata pesureseptejä. Myös tunnusvärin annostelu-aikaa ja annostelun ajankohtaa voidaan muuttaa reseptisivulta. Hälytykset-sivu näyttää aktiiviset hälytykset ja sivulta voidaan myös kuitata hälytykset. Trendisivut on tehty sekoitinsäiliöille, päiväsailiöille, pesuvesisäiliöille, hartsisäiliöille ja kovetesiilolle. Laitteille on myös laitekohtaiset käsinajosivut, jotka aukeavat laitetta klikkaamalla, kun sekoitin on asetettu käsinajolle.

5.2 Logiikan ja valvomon kommunikointi

Valvomo tietokoneella on Data Access Server (DAServer), jonka välityksellä InTouch ja logiikka kommunikoivat ethernet-protokollaa käyttäen. DAServerille asetetaan logiikan asetukset ja InTouchin Device Groupen nimet sekä niiden päivitysnopeudet. Tagia tehdessä valitaan access name, jolla asetetaan, minkä device groupin kautta kommunikointi tapahtuu. Tässä projektissa kaikilla device groupilla on sekunnin päivitysnopeus, joten valinta ei vaikuta päivitysnopeuteen.

5.2.1 Yhteyden valvonta

Valvomon ja logiikan välinen kommunikointi on valvomon toiminnan kannalta välttämätöntä toimia, joten yhteyden toiminnalle tehtiin valvonta ja hälytys. Valvontaa varten logiikkaan tehtiin laskuri, jonka arvoa kasvatetaan kerran sekunnissa yhdellä. Laskuri asetetaan nolaksi sen saavuttaessa arvon 65535, jonka jälkeen laskenta alkaa uudestaan nolasta. Valvomon application -scriptiin tehtiin alla oleva koodi yhteyden valvontaa varten. Scripti on asetettu jatkuvasti suoritettavaksi 1000 millisekunnin välein.

```
IF OldCommUpdateCounter <> LogiikkaYhteysvahti THEN
```

```

OldCommUpdateCounter = LogiikkaYhteysvahti;
CommErrorCounter = 0;
ELSE
IF OldCommUpdateCounter == LogiikkaYhteysvahti THEN
CommErrorCounter = CommErrorCounter + 1;
ENDIF;
ENDIF;

IF CommErrorCounter >= 30 THEN
CommError = 1;
ELSE
CommError = 0;
ENDIF;

IF CommError == 1 THEN
ShowöKkommunikointivirheö;
ENDIF

```

LogiikkaYhteysvahti on logiikalta tulevan laskurin arvo. Ensimmäisellä rivillä verrataan, onko LogiikkaYhteysvahti arvo erisuuri kuin OldCommUpdateCounter. Jos ensimmäisen rivin ehto toteutuu, toisella rivillä asetetaan OldCommUpdateCounter arvoksi LogiikkaYhteysvahti arvo, ja kolmannella rivillä asetetaan nolaksi CommErrorCounter. Jos ensimmäisen rivin ehto ei toteudu, viidennellä rivillä varmistetaan, että OldCommUpdateCounter on yhtäsuuri kuin LogiikkaYhteysvahti. Mikäli ehto toteutuu, kuudennella rivillä CommErrorCounter arvoon lisätään yksi. Seuraavaksi verrataan, onko CommErrorCounter suurempi tai yhtäsuuri kuin 30. Jos ehto toteutuu, asetetaan CommError arvoksi 1. Jos ehto ei toteudu, arvoksi asetetaan 0. CommError arvon ollessa 1 näytetään kommunikointivirhe -sivu. Ohjelma suoritetaan kerran sekunnissa, joten kommunikointivirhe -sivu avataan, jos LogiikkaYhteysvahti arvo ei ole muuttunut 30 sekuntiin.

5.2.2 Käytetyt datatyypit

Valvomon ohjelmissa tagia tehdessä määritellään datatyyppi. Käytettyjä datatyyppejä ovat sisäisesti käytettävissä olevat Memomy-tyyppiset sekä ulkoisen järjestelmän kanssa yhteydessä olevat I/O-tyyppiset. Tagia määriteltäessä valitaan oikea datatyyppi, ja jos kyseessä on logiikan kanssa yhteydessä oleva tagi, määritellään tagiin myös logiikan osoite, josta arvo luetaan tai mihin arvo kirjoitetaan. Datatyypit jakautuvat Discrete-, Integer-, Real- ja Message-tyypisiin.

Discrete-tyyppisellä muuttujalla on vain kaksi mahdollista tilaa, 0 tai 1. Logiikan puolella tämä vastaa bool-tyyppistä muuttujaa, jonka osoite voi olla esimerkiksi DB300.DBX0.0.

Integer-tyyppisen muuttujan arvo voi olla kokonaisluku väliltä -2 147 483 648 ó 2 147 483 647. Logiikassa tämä vastaa integer-tyyppistä muuttujaa, jonka osoite voi olla esimerkiksi DB300.DBW2.

Real-tyyppisen muuttujan arvo voi olla desimaaliluku väliltä $-3,4e^{38}$ ó $3,4e^{38}$. Logiikassa tämä vastaa real-tyyppistä muuttujaa, jonka osoite voi olla esimerkiksi BD300.DBD4.

Message-tyyppinen muuttuja voi olla maksimissaan 131 merkin pituinen merkkijono. Logiikassa tämä vastaa string-tyyppistä muuttujaa, jonka osoite voi olla esimerkiksi DB300.STRING[32].310.

5.3 Tilatiedot logiikalta valvomolle

Päätin tuoda tilatiedot valvomolle logiikan data blockien kautta. Esimerkiksi logiikan tulot olisi voinut lukea suoraan valvomolle, mutta mielestäni oli selkeämpää tuoda myös ne data blockin kautta. Tein tilatietoja varten blockit FC300 (function) ja DB300 (data block). FC300:ssa siirretään tarvittavat tilatiedot DB300:een. FC300:ssa suoritetaan myös antureiden skaalauksia, jotta anturien mittaamat arvot saadaan ymmärrettävään muotoon. Valvomo lukee tilatiedot DB300:sta.

5.3.1 Hälytykset

Liimakeittion hälytysten saamiseksi valvomoon käytin hyödyksi entuudestaan olleita data bloqueja DB22 ja DB23. DB22 sisältää prosessihälytykset vaakoja lukuunottamatta ja DB23 vaakojen hälytykset. Hälytykset oli tehty logiikkaan jo valmiiksi, joten tehtäväksi jäi vain hälytysten, hälytysten kuittauksen ja hälytysten indikoinnin teko valvomoon.

Jokaiselle hälytykselle täytyi tehdä valvomoon oma tagi. Tagin asetuksista oleellimmat ovat tagname, type, group, item ja alarm comment. Tagname on yksilöllinen nimi tagille. Type valitsee tagin tyypin. Tässä tapauksessa type on I/O discrete, eli logiikkaan yhteydessä oleva tagi, jonka arvo voi olla nolla tai yksi. Tagi on asetettu read only-tyyppiseksi, sillä aktiiviset hälytykset ainoastaan luetaan logiikalta. Group kohtaan valitaan Liimakeittiö. Kaikkien hälytysten group on sama. Myöhemmin toista sekoitinta käyttöön ottaessa on mahdollista tehdä myös toinen ryhmä, jolloin hälytysindikoinnit voidaan tarvittaessa eriyttää. Access-nameksi on asetettu Häiriöt. Alarm comment on teksti, joka lukee hälytys sivulla, kun hälytys on aktiivinen. Kuvassa 7 esimerkki hälytyksen tagista.

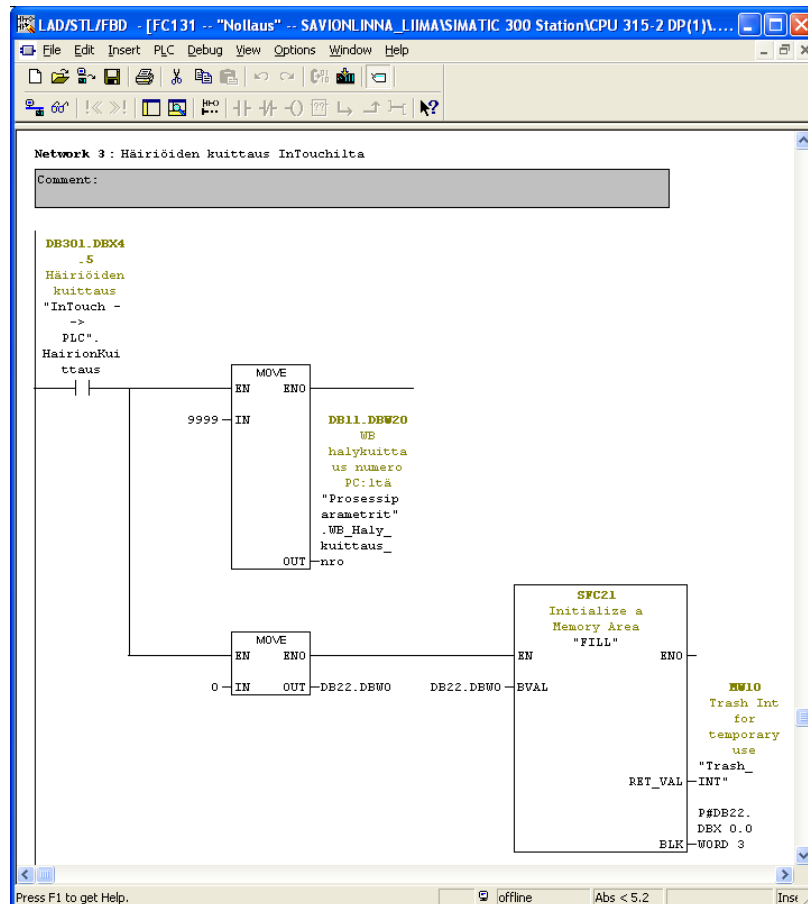
The screenshot shows the 'Tagname Dictionary' dialog box with the following configuration:

- Tags:** Main, Details, Alarms, **Details & Alarms**, Members
- Buttons:** New, Restore, Delete, Save, <<, Select..., >>, Cancel, Close
- Tagname:** Häl_V1_4100-M1EKäy
- Type:** I/O Discrete
- Group:** Liimakeittiö
- Read only:** ☒ Read only, ☐ Read/Write
- Comment:** (empty text box)
- Log Data:** ☐ Log Data, ☐ Log Events, ☐ Retentive Value
- Initial Value:** ☐ On, ☒ Off
- Input Conversion:** ☒ Direct, ☐ Reverse
- On Msg:** (empty text box), **Off Msg:** (empty text box)
- Access Name:** Häiriöt
- Item:** db22.x1.6, ☐ Use Tagname as Item Name
- ACK Model:** ☒ Condition, ☐ Event Oriented, ☐ Expanded Summary
- Alarm Comment:** Vaaka1: Säiliön sekoitin 4100-M1 ei käy
- Alarm State:** ☒ On, ☐ Off, ☐ None
- Priority:** 1
- Alarm Inhibitor:** (empty text box)

KUVA 7. Hälytystagi

Hälytysten kuittaukselle on painike hälytyssivulla. Painike kuittaa kaikki hälytykset kerralla. Logiikkaohjelmassa painike kuittaa hälytykset vaakan yksi osalta siirtämällä numeron 9999 paikkaan DB11.DBW20. Arvo 9999 kuittaa kaikki vaakan hälytykset. Tämä toiminta oli jo valmiiksi tehty blockissa FC116 PC_AlarmCLR_WB. Prosessihälytykset kuitataan siirtämällä 0 paikkaan DB22.DBW0. DB22.DBW0:n arvolla puolestaan alustetaan kaikki DB22:n arvot nolliksi. Alustamiseen käytetään Siemensin

valmista Initialize a Memory Area -toimintoa. Kuittauksen toteutus ilmenee kuvassa 8. BVAL arvoksi asetetaan arvo, jolla kohde halutaan alustaa, eli DB22.DBW0:n sisältö. Alustettava kohde ja alustettavan alueen laajuus asetetaan lähdön BLK arvolla. Tässä tapauksessa alustetaan kolme sanaa alkaen DB22.DBX0.0:sta.



KUVA 8. Hälytysten kuittaus logiikassa

Kun liimakeittiöllä on aktiivisia hälytyksiä, hälytykset-sivulle tulee lista aktiivisista hälytyksistä. Myös valvomon hälytykset sivun avaavan painikkeen vieressä oleva vihreä ympyrä muuttuu punaiseksi. Lisäksi logiikalle menee tieto aktiivisten hälytysten määrästä, jonka perusteella sytytetään häiriömerkkivalo sekä lähetetään tieto häiriöstä eteenpäin. Hälytysten indikointia varten InTouchin application-scriptiin on tehty seuraavat ohjelmarivit:

```
{Hälytyksien lkm}
```

```
Hälytyksetlkm = Liimakeittiö.AlarmTotalCount;
```

```
JokinHälytysPäälläPLC = Hälytyksetlkm;
```

```
IF Hälytyksetlkm > 0 THEN
```

```

JokinHälytysPäällä = 1;
ELSE
JokinHälytysPäällä = 0;

ENDIF;

```

Aluksi Hälytyksetlkm arvoksi asetetaan Liimakeittiö.AlarmTotalCount:n arvo. AlarmTotalCount on käsky, joka ilmaisee hälytysten lukumäärän tagista tai hälytysryhmästä. Tässä tapauksessa käskyllä saadaan Liimakeittiö öhälytysryhmän aktiivisten hälytysten määrä. Seuraavalla rivillä Hälytyksetlkm:n arvo asetetaan JokinHälytysPäälläPLC arvoksi. Tämän jälkeen asetetaan JokinHälytysPäällä tagin arvoksi yksi, jos Hälytyksetlkm arvo on suurempi kuin nolla. Jos ehto ei toteudu, asetetaan JokinHälytysPäällä arvoksi nolla. JokinHälytysPäällä ohjaa aiemmin mainittua valvomon häiriöindikointia. JokinHälytysPäälläPLC tagi on logiikassa DB301.DBW278. Jos DB301.DBW278 arvo on suurempi kuin 0, logiikan lähtö Q12.1 ohjataan päälle. Q12.1 ohjaa lähtörelettä, joka ohjaa häiriömerkkivaloa. Lähtöreleen rinnalla on myös toinen rele, jonka kautta saadaan tieto häiriöstä häiriötekstiviestejä lähettävään järjestelmään.

5.3.2 Sekoittimen tilatieto

Liiman komponenttien annostelu sekoittimelle tehdään logiikkaohjelmassa sekvenssiohjauksena. Jokaiselle sekvenssille on oma numeronsa, ja kun edellinen sekvenssi on saatu valmiiksi, siirrytään seuraavaan. Käytössä olevat sekvenssit ovat: 0 (Vapaa), 40 (Pesuveden annostelu), 50 (Veden annostelu), 60 (Kovete 1 annostelu), 70 (Kovete 2 annostelu), 80 (Hartsin annostelu), 110 (Esisekoitus), 120 (Sekoitus), 140 (Tyhjennys) ja 150 (Odottaa tyhjennystä). Käynnissä oleva sekvenssi haluttiin päänäytölle näkyviin sekoittimelle tilatiedoksi. Tätä varten luotiin TilaTxt- ja TilaTxtAnim-tagit, sekä lisättiin koodia Application-scriptiin.

Sekvenssinumeroa käytetään ohjelmassa liimakomponenttien annostelun ohjaamiseen. DB20.DBW0 sisältää sekvenssinumeron, jota käytetään ohjelman suorittamiseen. DB20.DBW0:n arvo siirretään FC300:ssa DB300.DBW44:ään. TilaTxtAnim sisältää logiikan DB300.DBW44:n arvon. TilaTxt-tagtiin asetetaan application-scriptissä määritelty teksti TilaTxtAnim-tagin sisältämän sekvenssinumeron perusteella. Tekstit ase-

tetaan yksinkertaisilla IF- ja THEN ólauseilla. Jokaiselle tilalle on tehty oma IF ólause. Esimerkiksi pesuveden annostelu teksti kirjoitetaan TilaTxt-tagiin seuraavasti.

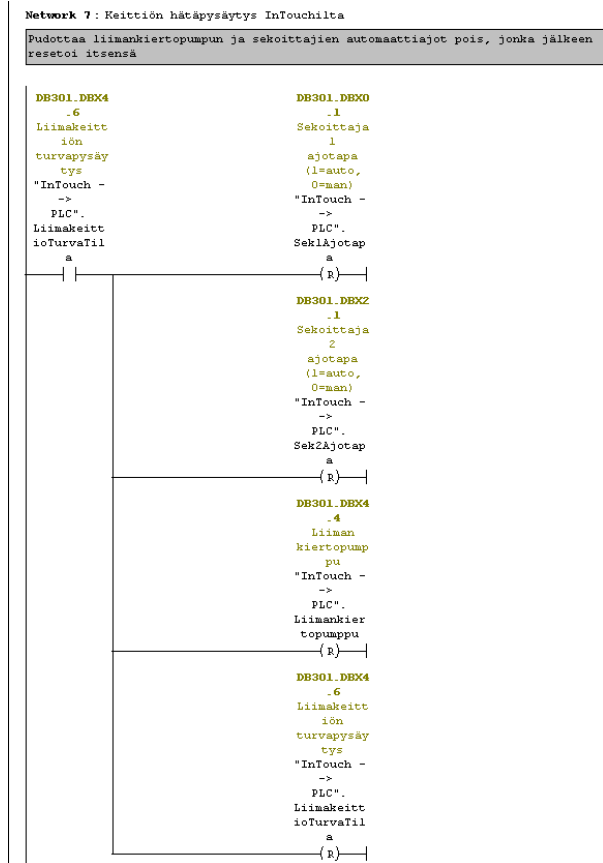
```
IF TilaTxtAnim == 40 THEN
  TilaTxt = "Pesuveden annostelu";
ENDIF;
```

Yllä verrataan ensin, onko TilaTxtAnim-arvo 40. Jos arvo on 40, TilaTxt:n sisällöksi asetetaan Pesuveden annostelu. Tämän jälkeen lopetetaan IF-lause.

5.4 Ohjaukset ja asetukset logiikalle

Ohjauksia ja asetuksia varten loin DB301:n ja FC301:n. Lisäksi entuudestaan oli tehty DB17 käsinajoja varten. Käsinajoihin liittyen ei tarvinnut tehdä muutoksia logiikkaohjelmaan, mutta valvomoon täytyi tehdä ohjaukset, jotka viittaavat oikeisiin bitteihin logiikassa.

Käsinajoja lukuunottamatta kaikki asetukset ja ohjaukset tuotiin DB301:een. Osaa DB301:n arvoista käytetään suoraan laitteita ohjaavissa function bloqueissa. Kaikkia arvoja ei kuitenkaan saanut järkevästi tuotua suoraan oikeanmuotoisena logiikkaan. Tällaiset arvot muokattiin sopiviksi FC301:ssä. Vaakojen materiaaliparametrit kierrätettiin FC301:n kautta, sillä useimmat parametriarvot piti muokata kymmenkertaisiksi. Myös pääsivulla oleva liimakeittiön hätäpysäytys toteutetaan FC301:ssä. Logiikan puolen toteutus kuvassa 9.



KUVA 9. Liimakeittion hätäpysäytyksen toteutus logiikassa

Pääsivulla oleva hätäpysäytys asettaa logiikasta bitin DB301.DBX4.6 päälle. Tämä bitti puolestaan asettaa kummankin sekoittajan ajotavat käsinajolle ja pysäyttää liiman kiertopumpun. Lopuksi bitti resetoidaan, jottei se jää pysyvästi ykköseksi.

5.5 Reseptit

Reseptien hallintaan käytin InTouchin omaa Recipe Manageria. Recipe Managerin ja reseptikäskyjen avulla valitaan, lisätään, poistetaan ja muokataan reseptejä. Reseptit tallennetaan valvomo tietokoneella oleviin csv-tiedostoihin. Sekoittajat eroavat tilavuuksiltaan, joten molemmille sekoittajille on omat reseptit ja csv-tiedostot. Ajo-reseptien lisäksi on sekoitinkohtaiset muokattavissa olevat pesureseptit, joita ei ole tehty Recipe Managerin avulla, vaan reseptit ovat ainoastaan logiikassa tallessa. Pesureseptien muokkaus ja reseptien näyttö ovat myös valvomon reseptit sivuilla. Pesureseptillä ajetaan sekoittajan puhdistus joko automaattisesti aseteltavissa olevan satsimäärän välein tai käsin tarvittaessa. Valvomosta siirretään haluttu resepti logiikkaan, joka ohjaa vaakayksikön avulla halutun määrän liimakomponentteja sekoittimeen sekä sekoittaa halutun ajan liimaa ennen pumppausta päivä säiliöön.

5.5.1 Reseptien hallinta InTouchissa

Recipe Managerilla luodaan resepti tiedostot. Reseptejä lisätään, poistetaan, muokataan ja valitaan resepti käskyillä. Kuvassa 10 olevalla reseptisivulla näkyy reseptien hallintaa varten tehty käyttöliittymä. Reseptin alapuolella olevat napit suorittavat edellä mainittuja resepti käskyjä. Pesuresepti, tunnusvärin annostelu sekä sekoittimen nopeus ovat I/O-tageja, jotka ovat yhteydessä logiikan data bloqueihin.

Resepti vaaka 1	
Reseptin nimi:	Säätö 1
Tallennusaika:	10:45:47 19.11.2014
Reseptin numero:	1
Kommentti:	Normaali
Pesuvesi (kg):	47
Harts (kg):	293
Raakavesi (kg):	63
Kovete (kg):	85
Harts (kg):	112
Esisekoitusaika (s):	360
Loppusekoitusaika (s):	120
Reseptin massa yhteensä (kg):	600
<input type="button" value="←"/> <input type="button" value="Hae resepti"/> <input type="button" value="→"/>	
<input type="button" value="Tallenna resepti"/> <input type="button" value="Poista resepti"/>	
<input type="button" value="Valitse resepti ajoon"/>	
<input type="button" value="Sulje ikkuna"/>	

Pesuvesiresepti	
Pesuvesi (kg):	540
Raakavesi (kg):	0
Sekoitusaika (s):	360
Sekoittimen nopeus (rpm):	1000
Tunnusvärin annostelu Annostelu aika (s): 15 Materiaaliluku, jonka aikana annostellaan: 3 1 = Pesuvesi 2 = Raakavesi 3 = Kovete 5 = Harts	
Sekoittimen nopeus (yhteinen kaikille resepteille) Sekoittimen nopeus (rpm): 1000	

KUVA 10. Reseptisivu

Poista resepti -nappi poistaa valittuna olevan reseptin suorittamalla action scriptin, jossa käytetään RecipeDelete-käskyä. Alla action scripti:

```
RecipeDelete(öc:\data\Liimakeittiö_SLN\liimaresepti.csv,      Reci-
peName);.
```

Tallenna resepti -nappi tallentaa reseptiin tehdyt muutokset tai uuden reseptin suorittamalla action scriptin, jossa asetetaan FenoliReseptiAika-muuttuja sekä suoritetaan RecipeSave-käsky. Alla action scripti:

```
FenoliReseptiAika = $TimeString + ö ö + $DateString;
```

```
RecipeSave(öc:\data\Liimakeittiö_SLN\liimaresepti.csvö, UnitName,
RecipeName );
```

Reseptin valintanuolet vaihtavat seuraavaan tai edelliseen reseptiin. Nuoli -napit suorittavat action scriptit, joissa suoritetaan RecipeSelectNextRecipe- tai RecipeSelectPreviousRecipe- sekä RecipeLoad-käskyt. Alla action scripti, jolla vaihdetaan edellinen resepti:

```
RecipeSelectPreviousRecipe(öc:\data\Liimakeittiö_SLN\
liimaresepti.csvö , RecipeName, 131 );
RecipeLoad( öc:\data\Liimakeittiö_SLN\liimaresepti.csvö, UnitName,
RecipeName);
```

Hae resepti -nappi avaa listan tallennetuista resepteistä, joista voidaan valita avattava resepti. Nappi suorittaa action scriptin, jossa suoritetaan RecipeSelectRecipe- ja RecipeLoad -käskyt. Alla action scripti, jolla valitaan resepti listasta:

```
RecipeSelectRecipe(öc:\data\Liimakeittiö_SLN\liimaresepti.csvö, Reci-
peName, 131 );
RecipeLoad(öc:\data\Liimakeittiö_SLN\liimaresepti.csvö, UnitName,
RecipeName, );
```

Valitse resepti ajoin -nappi siirtää näkyvissä olevan reseptin logiikalle, jolloin seuraava liimapanos tehdään uuden valinnan mukaisesti. Nappi suorittaa action scriptin, joka siirtää erikseen kaikki reseptin tiedot logiikkaan. Yhtäkuin merkkien oikealla puolella ovat memory-tyyppiset muuttujat, jotka ovat valvomon resepti sivulla olevat arvot. Vasemmalla puolella yhtäkuin merkkiä ovat I/O-tyyppiset muuttujat, jotka ovat yhteydessä logiikkaan. Alla action scripti, jolla siirretään resepti logiikkaan:

```
SFenoliReseptiNimi = RecipeName;
SFenoliReseptiAika = FenoliReseptiAika;
SFenoliReseptiKommentti = FenoliReseptiKommentti;
SFenoliReseptiNumero = FenoliReseptiNumero;
SFenoliReseptiHartsu = FenoliReseptiHartsu;
SFenoliReseptiRaakavesi = FenoliReseptiRaakavesi;
```

```

SFenoliReseptiPesuvesi = FenoliReseptiPesuvesi;
SFenoliReseptiKovete1 = FenoliReseptiKovete1;
SFenoliReseptiHarts2 = FenoliReseptiHarts2;
SFenoliReseptiEsisekoitusai = FenoliReseptiEsisekoitusai;
SFenoliReseptiLoppusekoitusai = FenoliReseptiLoppusekoitusai;

```

5.5.2 Reseptien käyttö logiikassa

Seuraavaksi ajoon haluttu resepti siirretään InTouchilta logiikan DB320:een. FC320:ssä reseptin tiedot siirretään jo olemassa olleeseen DB12:sta. Siirrettäviä tietoja ovat komponenttien materiaalinumerot, tavoitepainot, sekoitusajat, sekoittajan nopeus ja väriaineen annostelu aika. Materiaalinumero tulee InTouchin materiaali parametrit sivulta. Tavoitepaino on reseptissä aseteltu tavoitepaino kilogrammoina, joka täytyy kertoa kymmenellä. Sekoitusajat ovat sekunteina, kuten myös väriaineen annostelu aika. Väriaineen annostelu aika kerrotaan kymmenellä. DB320:een ja siitä edelleen DB12:sta siirretään varsinaisen reseptin lisäksi pesuresepti. Ajoresepti on DB:ssä DBW0:sta alkaen ja pesuresepti DBW200:sta alkaen. DB12:sta arvot näkyvät valvomon päänäytön taulukossa sarakkeessa seuraava resepti.

Reseptien käyttöön ohjelmassa liittyy olennaisesti kolme jo olemassa ollutta data blockia. DB12:sta on seuraavaksi ajoon tuleva resepti. DB21 on nimeltään resepti ajonäyttö, josta selviää ajossa oleva resepti sekä se, missä vaiheessa reseptin teko on. DB24:ää ei tarvita reseptin mukaisen satsin tekoa varten, vaan se on valvomolle menevää raporttia varten. Siitä näkee reseptin tavoite ja toteutuneet arvot. Sekoittimeen annostelua reseptin mukaan ohjataan FB43:ssa.

Joka kerta kun satsia aletaan sekoittamaan, siirretään DB12:sta sisältö DB21:een. Jos seuraava resepti on pesuresepti, siirretään DB21:een arvot DB12.DBW200:sta alkaen. Muuten arvot siirretään DB12.DBW0:sta alkaen. Ohjelma kirjoittaa tavoitteen lisäksi toteutuneet arvot DB21:een. Valvomon päänäytön taulukon sarakkeiden resepti ajossa ja toteutunut arvot tulevat DB21:stä. Kun satsi on valmis ja sekoitin tyhjennetty, siirretään DB21:n arvot vielä tiedonkeruuta varten DB24:ään. Reseptin mukana ei aiemmin siirretty reseptin numeroa ja nimeä, mutta tiedonkeruun vuoksi myös nämä tiedot piti saada kulkemaan reseptin mukana. Tämän vuoksi lisäsin ohjelmaan reseptin numeron ja nimen siirrot. Seuraavaksi ajoon tulevan reseptin numero ja nimi tulevat val-

vomosta DB320:een, kuten muutkin reseptitiedot. Kun resepti otetaan ajoin, eli DB12:sta sisältö siirretään DB21:een, siirretään myös numero DB320.DBD400:sta DB320.DBD438:aan ja nimi 404:stä 442:een. Tietojen siirtämisen tein Copying Memory Area -toiminnolla.

6 TIEDONKERUU

Tiedonkeruun tarkoituksena on kerätä tietoa tuotantolinjan toiminnasta. Kerättäviä tietoja voivat olla linjasta riippuen esimerkiksi linjalla ajatut juoksumetrit, kuutiot tai kappalemäärät. Usein kerätään myös hyötysuhteita ja laatujaumia. Tiedonkeruussa oleville tuotantolinjoille on tehty tiedonkeruumääritykset, joiden perusteella tiedonkeruu ohjelmoidaan. InTouchilla luodaan tiedonkeruuta varten tiedonkeruumääritysten mukainen csv-tiedosto. Csv-tiedoston sisältö luetaan serverille, joka luo tiedonkeruureportin.

6.1 Tiedonkeruumäärittelyt

Tiedonkeruumäärittelyissä määritellään, mitä tietoja, missä järjestyksessä ja missä muodossa csv-tiedostoon pitää kirjoittaa. Myös csv-tiedoston nimi ja polku määritellään tiedonkeruumäärittelyissä. Csv-tiedosto on ascii-tiedosto, jossa tietokentät on erotettu toisistaan pilkulla. Merkkijonot suljetaan lainausmerkkien sisään. Integer-tyyppiset muuttujat esitetään puolestaan ilman lainauksia. Liimakeittiöltä kerätään tiedot ainoastaan valmistetuista liimapanoksista. Liimakeittiöltä kerättävät tiedot näkyvät liitteessä 1 olevassa taulukossa. Taulukossa näkyvät kerättävien tietojen lisäksi raja-arvot ja otsikotietojen selitykset. Punaisella kirjoitetut eivät ole käytössä, sillä liimansekoittajissamme ei käytetä välisekoitusaikaa, eikä satsin tiheyttä mitata jokaisesta panoksesta.

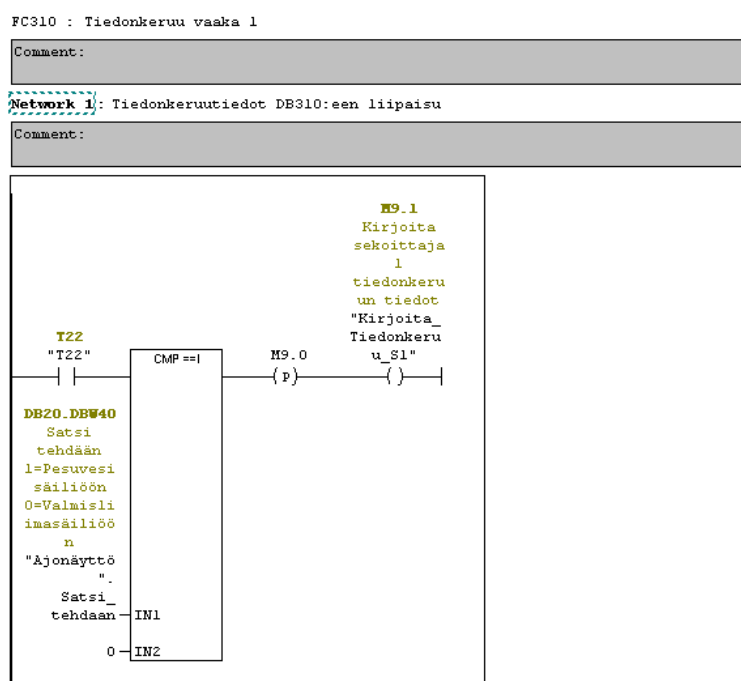
6.2 Tiedonkeruun ohjelmointi

Tiedonkeruuta varten täytyi tehdä ohjelmaa niin logiikkaan kuin InTouch-valvomoon. Valvomon tiedonkeruun ohjelmointiin oli jo valmis pohja, jota joutui muokkaamaan. Logiikan puolelle oli valmiina data block, johon tiedot kirjoitetaan, mutta tietojen ke-

räämiseen täytyi tehdä ohjelma. Lisäksi valvomon tageihin tuli määritellä oikeat logiikan osoitteet.

6.2.1 Ohjelmointi logiikassa

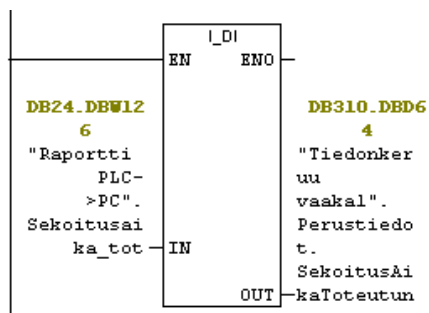
Logiikassa tiedonkeruun ohjelmointi on tehty blockissa FC310 ja tiedonkeruun arvot kirjoitetaan DB310:een. Jaottelin FC310:ssä tietojen kirjoittamiset seitsemään networkkiin. Ensimmäisessä tehdään tiedonkeruun kirjoituksen liipaisu. Tiedonkeruun tiedot kirjoitetaan uudestaan aina, kun satsi on tyhjennetty valmisliimasäiliöön. Panos valmis ja tyhjennetty sekvenssi asettaa Entended Pulse S5 Timerin yhdeksi sekunniksi päälle. Tämän lisäksi Compare Integer -vertailijalla varmistetaan, että satsi on tehty valmisliimasäiliöön. Kun nämä ehdot toteutuvat, Positive RLO Edge Detection liipaisee muistipaikan M9.1 yhden ohjelmakierron ajaksi päälle. Kirjoituksen liipaisu kuvassa 11.



KUVA 11. Tiedonkeruutietojen siirron liipaisu

M9.1-bittiä käytetään FC310:ssä tietojen siirtoon DB310:een. Toisessa networkissa siirretään perustiedot tiedonkeruuseen. Ensimmäisenä on tehty satsinumero laskuri, jonka arvoon lisätään yksi aina, kun satsi valmistuu. Kun muuttuja saavuttaa arvon 65535, siirretään arvoksi nolla. Arvo kirjoitetaan DB310.DBD0:aan, ja se on double integer -tyyppiä. Seuraavaksi kirjoitetaan integer-tyyppinen kohde DB310.DBW42:een. Kohdetietoa ei tällä hetkellä tarvita, joten sen arvoksi siirretään

aina arvo 1. Esisekoitus ja sekoitus aikojen tavoitteet ja toteutuneet arvot saadaan DB24:stä. Arvot täytyy tiedonkeruumääritysten vuoksi muuttaa integeristä double integeriksi. Tämä onnistuu managerin Integer to Double Integer -muuntimella. Kuvassa 12 toteutuneen sekoitusajan muuntaminen double integeriksi. Kuten kuvassa näkyy, integer-tyyppinen DB24.DBW126 luetaan tuloon ja lähtöön kirjoitetaan double integer -tyyppinen DB310.DBD64.

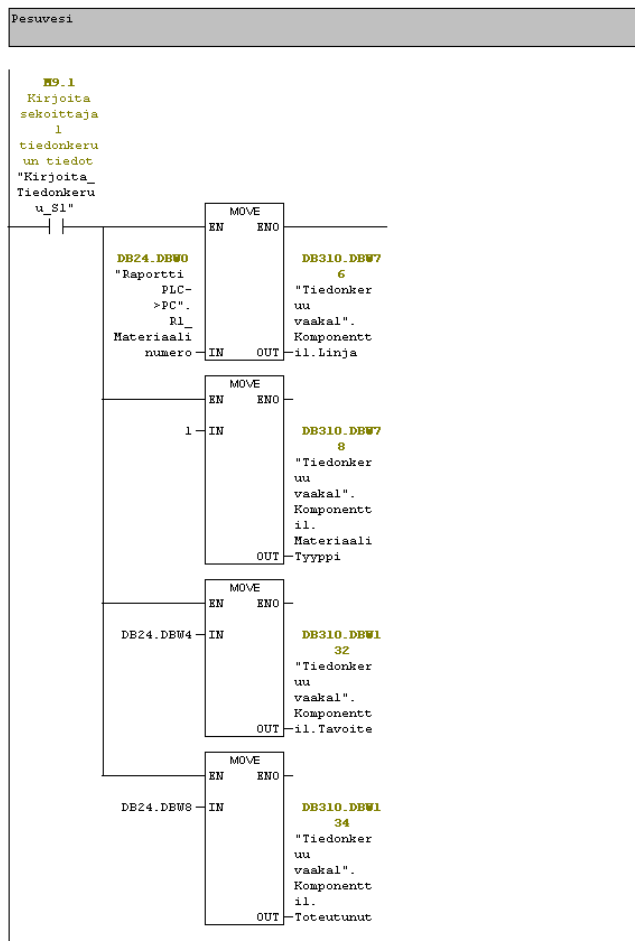


KUVA 12. Integer Double Integeriksi -muunnos

Myös reseptin numero ja nimi siirretään network kahdessa DB310:een. Numero ja nimi siirretään Copying Memory Area toiminnolla. Kirjoitin SRCBLK tuloon P#DB320.DBX438.0 WORD 19 ja lähtöön DSTBLK P#310.DBX4.0 WORD 19. Tällöin siirretään DB320.DBX438:sta alkaen 19 sanaa kohteeseen alkaen DB310.DBX4.0:sta. Kuvassa 13 perustietojen siirron network.

tavoite ja toteutuneet määrät. Tietojen siirto tehdään samalla pohjalla kaikilla käytössä olevilla komponenteilla, ainoastaan osoitteet ovat komponenttikohtaiset. Kuvassa 14 komponentin yksi tietojen siirto logiikkaohjelmassa DB310:een.

Network 3: Komponentin 1 tiedot tiedonkeruuseen



KUVA 14. Komponentin 1 tietojen siirto tiedonkeruuseen

Materiaalinumero saadaan siirrettyä suoraan DB24.DBW0:sta. Materiaalinumero on integer-tyyppinen muuttuja, ja se siirretään DB310.DBW76:een. Materiaalityyppi on kaikissa komponenteissa massa, eikä sitä ole tarvetta muuttaa, joten materiaalinumeron arvoksi siirretään yksi. Materiaalityyppi on myös integer-tyyppinen, ja se siirretään DB310.DBW78:aan. Komponenttien tavoite ja toteutunut määrien siirrossa muodostui ongelmaksi, että tiedonkeruuta varten tiedot tuli saada integer-tyyppisenä, mutta DB24:ssä ne ovat double integer -tyyppisiä. En löytänyt kyseistä muunnosta tekevää valmista toimintoa. Lopulta selvisi, että double integeristä voi lukea integer-luvun kahdesta jälkimmäisestä tavusta. Esimerkiksi DB24.DBD2:n arvon sai siirrettyä, kun move-piirin tuloon kirjoitti DB24.DBW4 ja lähtöön DB310.DBW132.

6.2.2 Ohjelmointi InTouchissa

Tiedonkeruuta varten tarvitaan valvomotietokoneelle csv tiedosto, josta tiedonkeruuserveri käy tiedot lukemassa. Tiedostoon kirjoitetaan tiedonkeruumääritysten mukaiset tiedot juuri oikeassa muodossa ja järjestyksessä. Tiedosto kirjoitetaan InTouchiin ohjelmoidun scriptin avulla. Scripti on QuickFunction -tyyppinen, ja sitä kutsutaan Application -scriptissä satsinumeron vaihtuessa. Jos BatchCsv1 on erisuuri kuin BatchCsv1_old, kasvatetaan Viive1-muuttujan arvoa yhdellä. Arvon ylittäessä neljä, kutsutaan BatchCsvWrite-scriptiä. BatchCsv1_old on tehty BatchCsvWrite-scriptissä. Alla koodi BatchCsvWrite-scriptin kutsusta.

```
IF BatchCsv1 <> BatchCsv1_old THEN
    Viive1 = Viive1 + 1;
    IF Viive1 > 4 THEN
        CALL BatchCsvWrite( );
    ENDIF;
ENDIF;
```

BatchCsvWrite-scriptin alussa luodaan message-tyyppiset BatchTemp-paikallismuuttujat. Tämä tapahtuu DIM-komennolla. DIM-komennon perään kirjoitetaan paikallismuuttujalle haluttu nimi, jonka jälkeen AS ja haluttu datatyyppi. Paikallismuuttujia voidaan käyttää ainoastaan scriptissä, jossa ne on luotu, eikä niiden arvo säily scriptin kirjoituksen jälkeen. Paikallismuuttujien luonnin jälkeen tekstiä sisältävät BatchCsv-muuttujat asetetaan välilyönneillä oikean pituisiksi. Tämän joutuu tekemään, sillä muuten csv-tiedostoon tulisi ainoastaan merkit, jotka on kirjoitettu, mutta tiedonkeruumääritykset vaativat tekstin tietyn mittaisena. Esimerkiksi BatchCsv3-muuttujaan lisätään 32 välilyöntiä (tyhjää merkkiä), jolloin muuttuja on pituudeltaan tiedonkeruumääritysten mukainen. Aaltosulkeissa oleva teksti on kommentti, joka ei vaikuta koodin toiminnallisuuteen. Alla koodi muuttujien valmistelusta.

```
{batch.csv tiedoston kirjoitus}
DIM BatchTemp1 AS MESSAGE;
DIM BatchTemp2 AS MESSAGE;
DIM BatchTemp3 AS MESSAGE;
DIM BatchTemp4 AS MESSAGE;
```

```

DIM BatchTemp5 AS MESSAGE;
DIM BatchTemp6 AS MESSAGE;
DIM BatchTemp7 AS MESSAGE;

```

```

BatchCsv3 = BatchCsv3 + "          ";
BatchCsv15 = BatchCsv15 + "        ";
BatchCsv16 = BatchCsv16 + "      ";
BatchCsv22 = BatchCsv22 + "      ";
BatchCsv23 = BatchCsv23 + "      ";
BatchCsv29 = BatchCsv29 + "      ";
BatchCsv30 = BatchCsv30 + "      ";
BatchCsv36 = BatchCsv36 + "      ";
BatchCsv37 = BatchCsv37 + "      ";

```

Kun muuttujat on alustettu, kerätään perustiedot, jotka asetetaan BatchTemp1:n arvoksi. Perustiedot ovat tiedonkeruumääritysten mukaisesti erotettu pilkuilla. Lisäksi merkkijonot suljetaan lainausmerkkien sisään. Lukuunottamatta reseptin nimeä (BatchCsv3), kaikki perustietojen muuttujat ovat numeroita. Nämä kirjoitetaan BatchTemp1:een Text-funktion avulla. Funktio muuttaa numeroarvon merkkijonoksi. Esimerkiksi Text(BatchCsv1, öö) luo merkkijonon, jonka arvona on BatchCsv1:n sisältämä numero. Pilkun jälkeisellä lainausmerkkien sisällä olevalla arvolla voidaan muuttaa esitettävän luvun muotoa, tässä tapauksessa luku esitetään normaalina kokonaislukuna. Perustietojen ainut merkkijono on BatchCsv3, joka sisältää reseptin nimen. Reseptin nimen molemmille puolille tehdään lainausmerkit StringChar-käskyllä. Käskyllä saadaan ASCII-koodin mukainen merkki. Tässä tapauksessa StringChar(34) lisää tekstiin lainausmerkit.

Kun tiedot on kirjoitettu BatchTemp1:n arvoksi, poistetaan vanha Batch.csv-tiedosto FileDelete-toiminnolla. FileWriteMessage käskyllä kirjoitetaan csv-tiedostoon BatchTemp1:n sisältö. Käskyn jälkeen annetaan suluissa tiedoston nimi, johon kirjoitetaan, kirjoituksen aloituskohta tavuissa, lähdetekstin tagin nimi sekä mahdollinen rivinvaihto tekstin jälkeen. Tässä tapauksessa kirjoitetaan Batch.csv-tiedostoon, kirjoitus aloitetaan tiedoston alusta, lähdetekstinä on BatchTemp1 ja kirjoituksen jälkeen ei tehdä rivinvaihtoa. Perustiedot kirjoitetaan csv-tiedostoon seuraavalla ohjelmakoodilla.

```

{Perustiedot}
BatchTemp1 = Text(BatchCsv1, "0") + "," + Text(BatchCsv2, "0") + ","
+ StringChar(34) + BatchCsv3 + StringChar(34) + "," +
Text(BatchCsv4, "0") + "," + Text(BatchCsv5, "0") + "," +
Text(BatchCsv6, "0") + "," + Text(BatchCsv7, "0") + "," +
Text(BatchCsv8, "0") + "," + Text(BatchCsv9, "0") + "," ;
BatchTemp1 = BatchTemp1 + Text(BatchCsv10, "0") + "," +
Text(BatchCsv11, "0") + "," + Text(BatchCsv12, "0") + ",";
FileDelete( "c:\rw\rundata\Batch.csv" );
FileWriteMessage( "c:\rw\rundata\Batch.csv", 0, BatchTemp1, 0 );

```

Perustietojen jälkeen kirjoitetaan komponenttien tiedot. Komponentit yhdestä neljään ovat käytössä, ja niiden kirjoitus tapahtuu keskenään samalla tavalla. Komponentin yksi kirjoitus tapahtuu muuten vastaavasti kuin perustietojen, mutta tiedot kerätään BatchTemp2⁶paikallismuuttujaan, tiedostoa ei poisteta FileDelete⁶käskyllä ja FileWriteMessage⁶-käskyllä kirjoitetaan tiedot Batch.csv-tiedoston loppuun. Tiedoston loppuun kirjoitus tehdään asettamalla kirjoituksen aloituskohdaksi -1. Komponentin yksi tietojen kirjoitus csv-tiedostoon tehdään seuraavalla ohjelmakoodilla.

```

{Komponentti 1}
BatchTemp2 = Text(BatchCsv13, "0") + "," + Text(BatchCsv14, "0") +
"," + StringChar(34) + BatchCsv15 + StringChar(34) + "," + String-
Char(34) + BatchCsv16 + StringChar(34) + "," + Text(BatchCsv17,
"0") + "," + Text(BatchCsv18, "0") + "," + Text(BatchCsv19, "0") +
",";
FileWriteMessage( "c:\rw\rundata\Batch.csv", -1, BatchTemp2, 0 );

```

Komponentit viidestä 20:een eivät ole käytössä. Jotta csv-tiedostosta saadaan oikean mittainen ja muotoinen, täytyy myös näiden komponenttien tiedot kirjoittaa. Tiedot kirjoitetaan ensin BatchTemp7⁷-paikallismuuttujan. Numerokentät asetetaan nolliksi ja tekstikenttiin asetetaan tiedonkeruumääritysten mukainen määrä välilyöntejä. Kun BatchTemp7 on kirjoitettu, kirjoitetaan sen sisältö FileWriteMessage⁶-käskyllä erikseen lopuille komponenteille. Alla koodi BatchTemp7⁶paikallismuuttujan asettamisesta ja komponentin viisi kirjoittamisesta csv-tiedostoon.

```

{Komponentti 5 - 20 ei käytössä}
BatchTemp7 = "0,0," + StringChar(34) + "          " + StringChar(34)
+ "," + StringChar(34) + "          " + StringChar(34) +
",0,0,0,";
{Komp 5}
FileWriteMessage( "c:\rw\rundata\Batch.csv", -1, BatchTemp7, 0 );

```

Lopuksi vielä asetetaan BatchCsvWrite-scriptin kutsumiseen käytettävät Viive1- ja BatchCsv1_old-tagit. Viive1 asetetaan nolaksi ja BatchCsv1_old-arvoksi asetetaan BatchCsv1:n arvo. Alla koodi Viiveen nollaamisesta ja BatchCsv1_old:n asettamisesta.

```

Viive1 = 0;
BatchCsv1_old = BatchCsv1;

```

6.3 Tiedonkeruuraaportti

Liimakeittiöstä tehdään kolme erillistä tiedonkeruuraaporttia. Ne ovat summaraportti, eräsummaraportti ja liimapanosraportti. Raportit voi luoda halutulta ajalta. Summaraportista näkyy materiaaliakohtaisesti asetetut ja toteutuneet annostelumäärät. Kuvassa 15 summaraportti ajalta 16.2.2015 klo.5.45 ó 20.2.2015 klo. 5.45.

UPM

SUMMARAPORTTI: 1190136003 - LIIMAKEITTIÖ

Raportointiväli 2015-02-16 05:45:00 - 2015-02-20 05:44:59
 Operaattori ?, ?
 Tulostusaika 23.2.2015 17:21:15
 Hakuehdot Operaattori=*, Ryhmä=*, Erätunniste=*, Ajotunniste=*

MATER. KOODI	MATERIAALI	MÄÄRÄ (ASET.)	MÄÄRÄ (TOT.)	MATER. TYYPPI
01	Pesuvesi	6498	6440,2	Kg
02	Harts	47379	47383,2	Kg
03	Raakavesi	6032	6033,4	Kg
04	Kovete	10290	10276,4	Kg

KUVA 15. Tiedonkeruun summaraportti

Eräsummaraportti on muuten vastaava kuin summaraportti, mutta siinä on annostelumäärät eroteltuina eri resepteillä. Raportissa näkyy täten reseptin nimi ja numero. Raportissa on myös kohde- ja linjarivit. Linjarivi on käytännössä turha, mutta kohdetietoa on tarkoitus käyttää myöhemmin sekoittimien yksilöintiin. Kuvassa 16 eräsummaraportti ajalta 16.2.2015 klo. 5.45 ó 20.2.2015 klo. 5.45.

UPM

ERÄSUMMARAPORTTI: 1190136003 - LIIMAKEITTIÖ

Raportointiväli 2015-02-16 05:45:00 - 2015-02-20 05:44:59
 Operaattori ?, ?
 Tulostusaika 23.2.2015 17:06:44
 Hakuehdot Operaattori=*, Ryhmä=*, Erätunniste=*, Ajotunniste=*

RES. NO	RESEPTIN NIMI	KOHDE	ERÄN STATUS	LINJA	MATER. KOODI	MATERIAALI	MÄÄRÄ (ASET.)	MÄÄRÄ (TOT.)	MATER. TYYPPI	KOMPON. STATUS
3	Säätö3	1	Ok	1	01	Pesuvesi	118	118	Kg	Ok
				2	03	Raakavesi	0	0	Kg	Ok
				3	04	Kovete	82	81,6	Kg	Ok
				5	02	Harts	399	398,8	Kg	Ok
4	Tavoite	1	Ok	1	01	Pesuvesi	6380	6322,2	Kg	Ok
				2	03	Raakavesi	6032	6033,4	Kg	Ok
				3	04	Kovete	10208	10194,8	Kg	Ok
				5	02	Harts	46980	46984,4	Kg	Ok

KUVA 16. Tiedonkeruun eräsummaraportti

Liimapanosraportti on raporteista tarkin. Raportista selviää jokaisen valmistetun liimapanoksen tiedot. Perustietoja ovat panoksen valmistumisaika, erä, reseptin numero, reseptin nimi, kohde, esisekoitusaika-asetus, esisekoitusaika toteutunut, sekoitusaika-asetus, sekoitusaika toteutunut ja erän status. Lisäksi on komponenttikohtaisia tietoja,

jotka ovat linja, materiaalikoodi, materiaali, määrä (asetus), määrä (toteutunut), materiaalityyppi, sekä komponentin status. Kuvassa 17 liimapanosraportti ajalta 20.2.2015 klo. 3.00 ó 20.2.2015 klo. 5.45.

http://153srv30/reports.asp?Starttime=2015-02-20 03:00:00&Endtime=2015-02-20 05:44:59&REPORTNAM - Windows Int...

UPM

LIIMAPANOSRAPORTTI: 1190136003 - LIIMAKEITTIÖ

Raportointiväli 2015-02-20 03:00:00 - 2015-02-20 05:44:59
 Operaattori ?, ?
 Tulostusaika 14.3.2015 14:52:49
 Hakuehdot Operaattori=*, Rytymä=*, Erätunniste=*, Ajotunniste=*

Paivita

AIKA	ERÄ	RES. NO	RESEPTIN NIMI	KOHDE	ESISEK. AIKA ASETUS sec	ESISEK. AIKA TOIEUT. sec	SEK. AIKA ASETUS sec	SEK. AIKA TOIEUT. sec	ERÄN STATUS	LINJA	MATER. KOODI	MATERIAALI	MÄÄRÄ (ASET.)	MÄÄRÄ (TOT.)	MATER. TYYPPI	KOMPON. STATUS
20.2.2015 3:03:01	1419	4	Tavoite	1	360	360	120	120	Ok	1	01	Pesuvesi	55	55,2	Kg	Ok
									Ok	5	02	Harts	405	404,8	Kg	Ok
									Ok	2	03	Raakavesi	52	52,2	Kg	Ok
									Ok	3	04	Kovete	88	87,6	Kg	Ok
20.2.2015 3:47:53	1420	4	Tavoite	1	360	360	120	120	Ok	1	01	Pesuvesi	55	55	Kg	Ok
									Ok	5	02	Harts	405	404,8	Kg	Ok
									Ok	2	03	Raakavesi	52	52	Kg	Ok
									Ok	3	04	Kovete	88	88	Kg	Ok
20.2.2015 4:49:01	1421	4	Tavoite	1	360	360	120	120	Ok	1	01	Pesuvesi	55	55,2	Kg	Ok
									Ok	5	02	Harts	405	405,2	Kg	Ok
									Ok	2	03	Raakavesi	52	52	Kg	Ok
									Ok	3	04	Kovete	88	87,6	Kg	Ok
20.2.2015 5:31:02	1422	3	Säätö3	1	360	360	120	120	Ok	1	01	Pesuvesi	118	118	Kg	Ok
									Ok	5	02	Harts	399	398,8	Kg	Ok
									Ok	2	03	Raakavesi	0	0	Kg	Ok
									Ok	3	04	Kovete	82	81,6	Kg	Ok

KUVA 17. Tiedonkeruun liimapanosraportti

Eri raportit soveltuvat eri tarkoituksiin. Summaraportista näkee nopeasti, kuinka paljon mitäkin materiaalia on kulunut määritellyssä ajassa. Eräsummaraportista puolestaan näkee, kuinka paljon milläkin reseptillä on ajettu. Liimapanosraporttia voi käyttää esimerkiksi materiaalin annostustarkkuuden seurantaan.

7 YHTEENVETO

Liimakeittiön tiedonkeruun teko ja valvomon uusinta onnistui hyvin. Työn aikataulu ei ollut käyttöönottoa lukuunottamatta tiukka, joten valvomon ja logiikkaohjelman sai suunnitella rauhassa valmiiksi. Valvomon layoutin suunnitteluun meni huomattavasti ennakoitua enemmän aikaa. Valvomon käyttöönoton tuli tapahtua sujuvasti, sillä muutos tehtiin tuotannon käydessä. Tämän vuoksi liimakeittiöltä täytyi saada jatkuvasti liimaa vanerin ladontaan. Valvomon käyttöönotto sujui ilman suuria ongelmia suunnittelussa aikataulussa. Valvomoa käyttävät pääasiassa tuotannon työntekijät, ja heiltä saatu palaute uudesta valvomosta on ollut positiivista. Uusitun valvomon layout vastaa todellisuutta ja käyttöliittymään ollaan oltu tyytyväisiä. Täten liimakeittiön käyttäminen on helpompaa kuin vanhalla valvomolla.

Tiedonkeruu otettiin käyttöön noin kuukausi valvomon käyttöönoton jälkeen. Myös tiedonkeruun ohjelmointi ja käyttöönotto sujui ilman suuria ongelmia. Tietojen kirjoitus csv-tiedostoon oikeassa muodossa vaati vähän ohjelman korjausta. Lisäksi pari viikkoa käyttöönoton jälkeen huomattiin, että säiliön pesuohjelman tiedot kirjoitettiin satunnaisesti tiedonkeruuseen. Vika ilmeni, kun heti pesuohjelman jälkeen aloitettiin liimapanoksen valmistus. Vika saatiin korjattua viivästäällä tiedonkeruutietojen kirjoitusta yhdellä sekunnilla logiikan ajastimen avulla.

Opinnäytetyön tarkoituksena oli saada tiedonkeruu sekoittajalle yksi. Tulevaisuudessa myös toinen sekoittaja on tarkoitus saada tiedonkeruuseen sekä uuteen valvomoon. Tätä varten sekoittajan vaakayksikkö täytyy vaihtaa ja ohjaukset siirtää nykyiseltä vaakayksiköltä logiikalle. Kun sekoittimen ohjaukset on saatu siirrettyä logiikalle, tiedonkeruun voi tehdä samalle pohjalle kuin sekoittajalla yksi. Suunnitteilla on myös automatisoida reseptien vaihto pesuvesimäärän mukaan, jolloin käyttäjien ei tarvitsisi vaihtaa käsin reseptejä pesuvesisäiliön pinnankorkeuden mukaan.

LÄHTEET

1. UPM. Vuosikertomus 2013. Verkkodokumentti.
http://www.upm.com/FI/SIJOITTAJAT/Documents/UPM_Vuosikertomus_2013.pdf. Muokattu 28.4.2014. Luettu 20.6.2014
2. UPM. Historia. Verkkodokumentti. <http://www.upm.com/FI/UPM/UPM-Lyhyesti/Historia/Pages/default.aspx>. Muokattu 10.11.2014. Luettu 10.11.2014
3. UPM. UPM Savonlinnan vaneritehdas. Verkkodokumentti.
<http://www.wisaplywood.com/fi/yhteystiedot/tuotantolaitokset/savonlinna/Pages/default.aspx>. Muokattu 11.3.2014. Luettu 11.3.2014.
4. Klinkmann. Wonderware. Verkkodokumentti.
<http://www.klinkmann.fi/tuotteet/automaatio/wonderware/>. Ei päivitystietoa.
Luettu 4.2.2015

1 LIIMAKEITTIÖN TIEDONKERUU

1.1 Liimapanos (liimakeittiö => tiedonkeruu)

Liimakeittiöstä tiedonkeruulle kerätään ainoastaan valmistettujen liimapanosten tiedot. Yhden liimapanoksen valmistus kestää useita minuutteja (n. 20 min). Näin ollen panostietojen puskurointia ei tarvita. Tiedonkeruujärjestelmä noutaa liimapanoksien tiedot liimakeittiön ohjausjärjestelmään kuuluvan PC:n kiintolevyltä tiedostosta

öc:\rw\rundata\batch.csvö. Em. tiedosto on ascii-tiedosto, jossa tietoken-
tät on erotettu toisistaan pilkuilla. Merkkijonot suljetaan lainausmerk-
kien sisään (esim. önäin ö), sitä vastoin integer -tyypin muuttujat esite-
tään ilman lainauksia (esim. 17294).

öbatch.csvö -tiedoston sisältämät kentät selviävät oheisesta taulukosta. Taulukon alussa luetellaan liimapanoksen otsikkotiedot, jonka jälkeen on varattu tilaa maksimissaan 20:lle panokseen sekoitettavalle komponentille. Käyttämättömien komponenttikenttien tiedot alustetaan seuraavasti: string:täytetään välilyönneillä, integer:0.

	Lähdeosoite	Raja-arvot ja dimensio	Selitys
Panoksen otsikkotiedot			
Satsinnumero		0..65535	ympäri pyörivä laskuri; kasvatetaan yhdellä ylöspäin uuden liimapanoksen talletuksen yhteydessä
ReseptiNumero		0..65535	
ReseptiNimi		char 32	
Kohde		1,2,3,4,...,20	kohdesäiliö
EsisekoitusAikaTavoite		0..65535 sec	
EsisekoitusAikaToteutunut		0..65535 sec	
VälisekoitusAikaTavoite		0..65535 sec	
VälisekoitusAikaToteutunut		0..65535 sec	
SekoitusAikaTavoite		0..65535 sec	
SekoitusAikaToteutunut		0..65535 sec	
SatsinTiheys		0..65535 g/l	Käsin syötettävä mittaustulos
PanosStatus		0:ok 1..65535:error	Panoksen valmistuksen onnistumisstatus
<<Komponenttitiedot >> [20]			
Linja		1,2,3,4,...,20	komponentin syöttölinjan nro
MateriaaliTyyppi		1,2	1:g (massa), 2:sec (aika)
MateriaaliKoodi		char 16	
MateriaaliNimi		char 32	
Tavoite		0..10000 x 0,1 kg 0..9999 x 0,1 sec	MateriaaliTyyppi=1 MateriaaliTyyppi=2
Toteutunut		0..10000 x 0,1 kg 0..9999 x 0,1 sec	MateriaaliTyyppi=1 MateriaaliTyyppi=2
Status		0:ok 1..65535:error	Komponentin annostuksen onnistumisstatus